

Digital *COSANDO*

Renzo Davoli

ALMA MATER STUDIORUM
Università di Bologna



COSARE

- Neologismo: costruire oggetti, cosi, bagagli, affari, robi, aggeggi, ... per il gusto di imparare
- Sono *COSI* digitali, che cioè sfruttano la semplicità a costruire *COSI* generata dal poter usare molto *software* e poco *hardware*.



COSE da adulti-bambini

- È un gioco (da bambini)
- Si usano tutti gli strumenti professionali (da adulti)
 - LINUX (mica Windows, MacOS, iOS o Android)
 - Linguaggi testuali (C, C++ o Python, mica scratch)



COMPLESSO

≠

COMPLICATO



What is a processor?

Cosa è un elaboratore?

- Computer è un termine obsoleto.
- Dove sono gli elaboratori?



Interruttore Luce Scale (schema 1984)

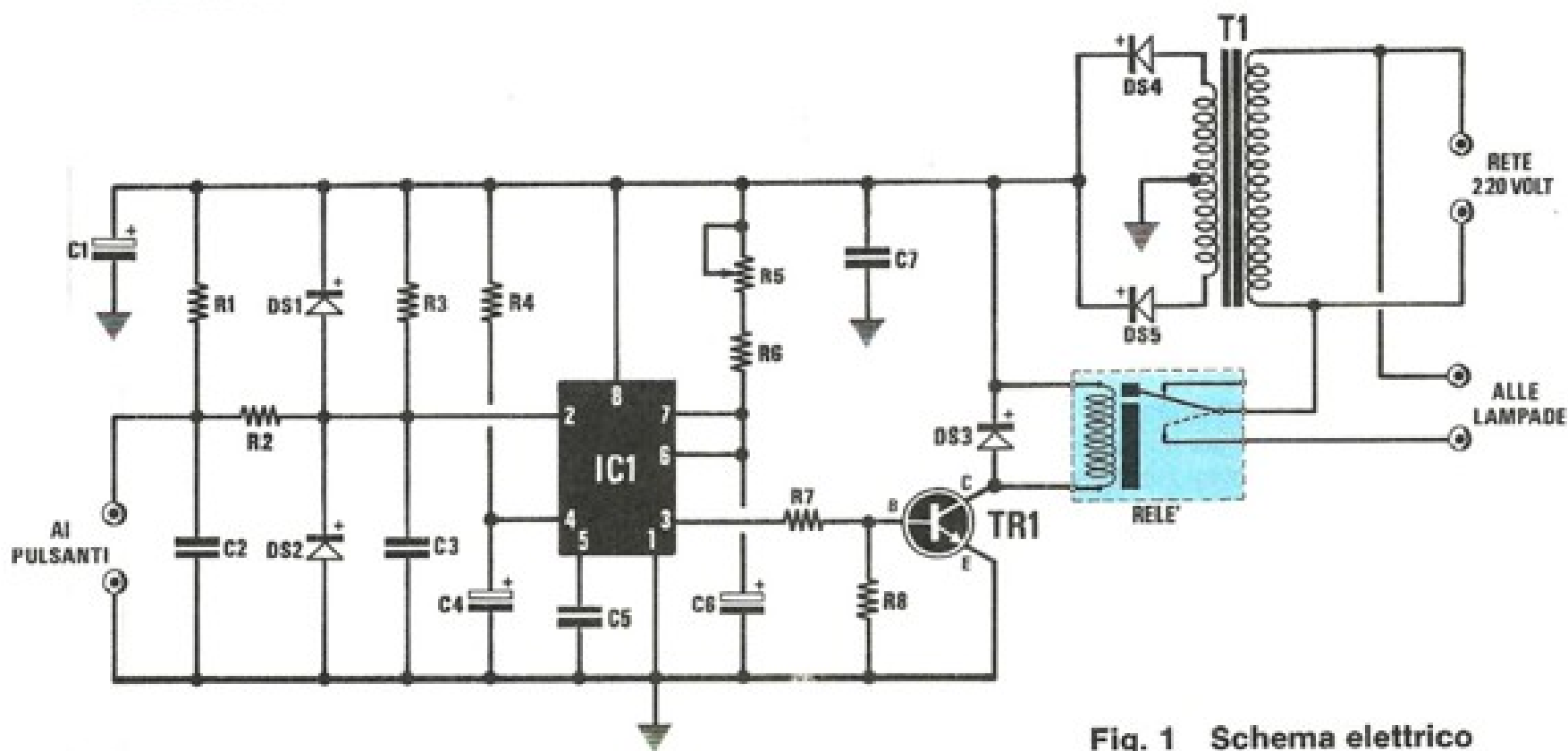
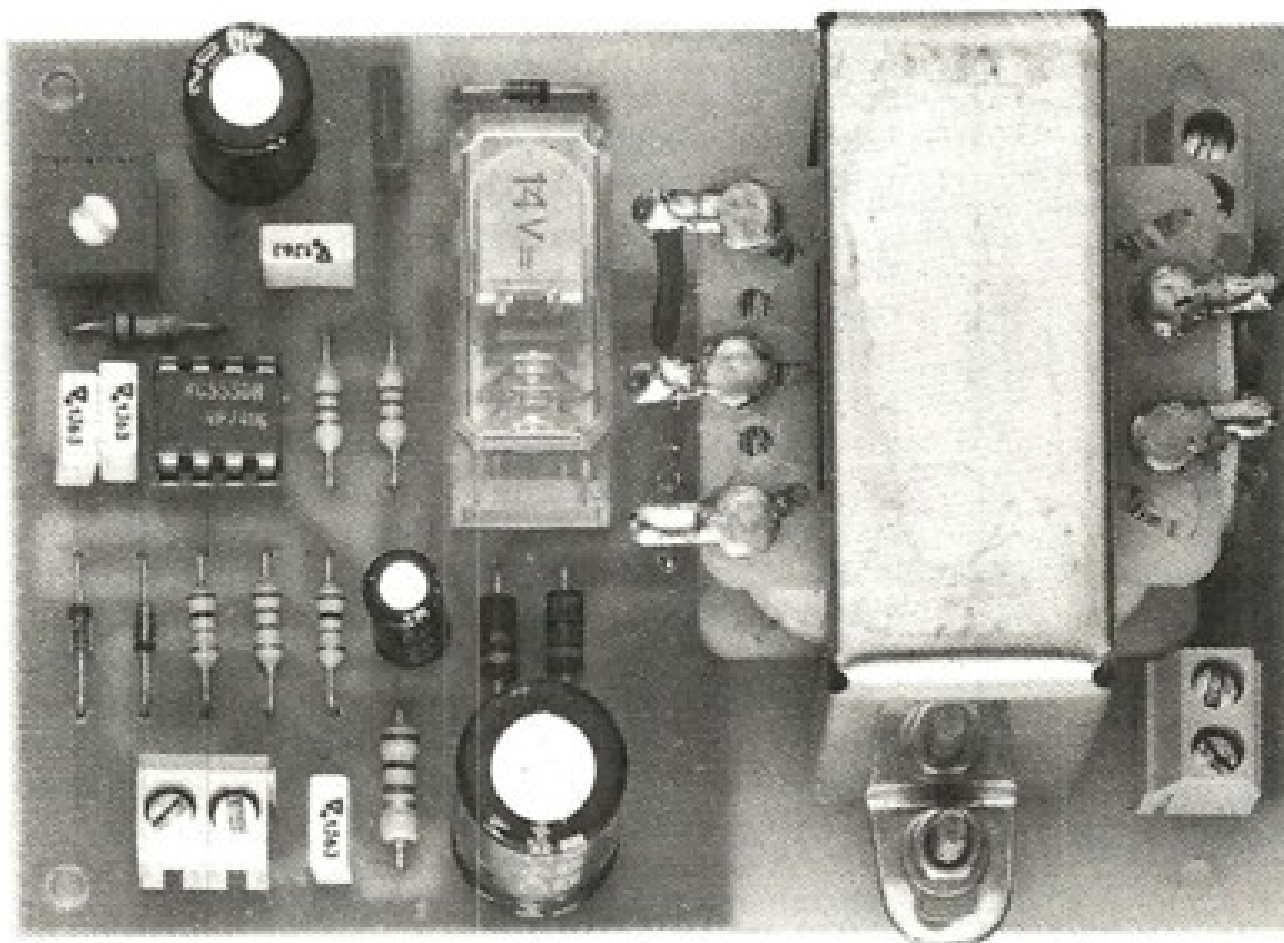
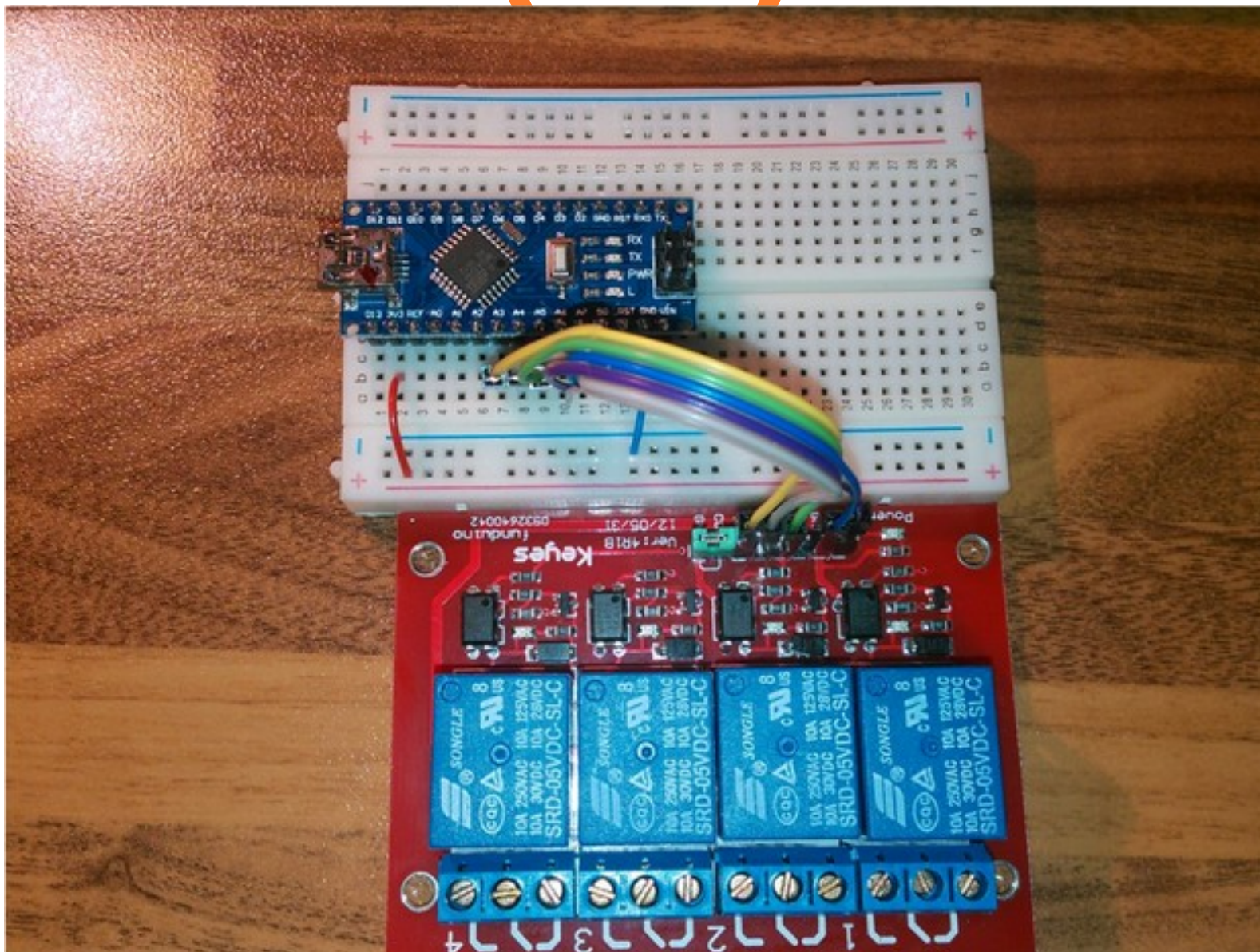


Fig. 1 Schema elettrico

Interruttore Luce Scale (schema 1984)



Interruttore Luce Scale (2018)



1984 vs 2018

- Tanti componenti
 - Circuito Stampato da saldare
 - Per Elettronici
 - Serve SOLO a quello scopo
 - (altrimenti occorre modificare il circuito)
 - Anche solo per modificare la durata occorre cambiare alcuni componenti.
 - MOLTO HW POCO SW
- Poche schede standard
 - No saldatore
 - Per Informatici +
 - Basta cambiare il software per farlo diventare:
 - un'intermittenza per luminarie
 - un simulatore di presenza
 - MOLTO SW POCO HW



Tutorial di Oggi

- Impareremo:
 - A far lampeggiare un LED
 - A costruire un interruttore per la luce delle scale
 - A leggere la temperatura
 - A costruire un termostato
- Abbiamo un sacco di cose da fare!
- Let's START!



Iniziamo a fare LOGIN

- Usando le credenziali
progettofermi.nnn@unibo.it
- OKAY?



Ora apriamo l'interfaccia per veri hacker: la shell!

- Occorre aprire un terminale
- Come si fa?
 - Hint: cercate terminal
- Dovreste ottenere una finestra con un messaggio di prompt simile a:

```
progettofermi.nnn@xxxxx:~$
```

 - (il vostro numero invece di nnn e il nome della macchina al posto di xxxxx)
- OKAY?



LA SHELL

- La shell si aspetta che voi digitiate dei comandi...
- Il comando che dovete digitare ora è
`/public/arduino-1.8.5/arduino`
- Dovrebbe apparire la finestra dell'IDE di Arduino...
- OKAY?
- Ma cosa è l'IDE e cosa ARDUINO?

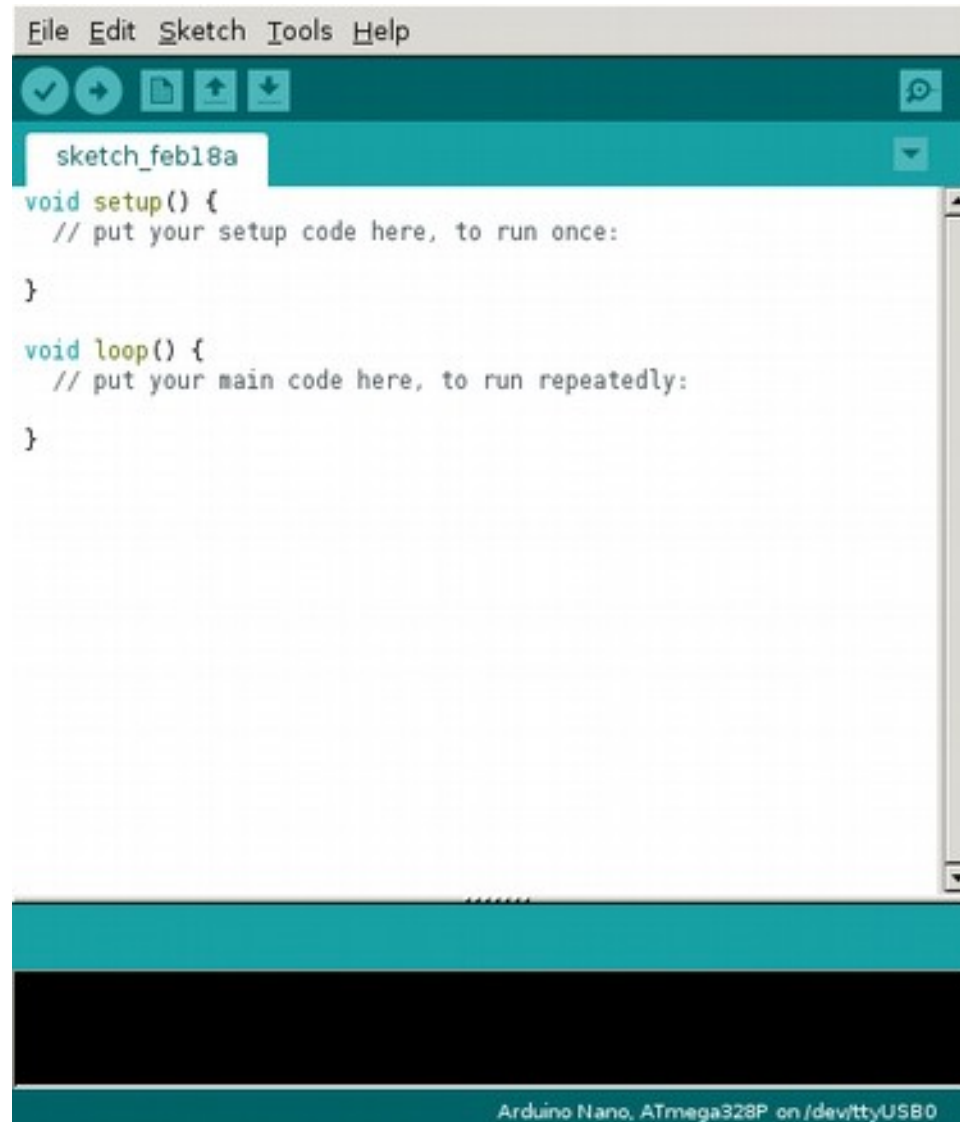


Integrated Development Environment

- In informatica un ambiente di sviluppo integrato (ovvero IDE) è un software che, in fase di programmazione, aiuta i programmatori nello sviluppo del codice sorgente di un programma. Spesso l'IDE aiuta lo sviluppatore segnalando errori di sintassi del codice direttamente in fase di scrittura, oltre a tutta una serie di strumenti e funzionalità di supporto alla fase di sviluppo e debugging.
- (definizione Wikipedia)



Ecco come appare l'IDE di ARDUINO



Ecco ARDUINO (alcuni componenti della sua grande famiglia)



Mr. Arduino

Massimo Banzi

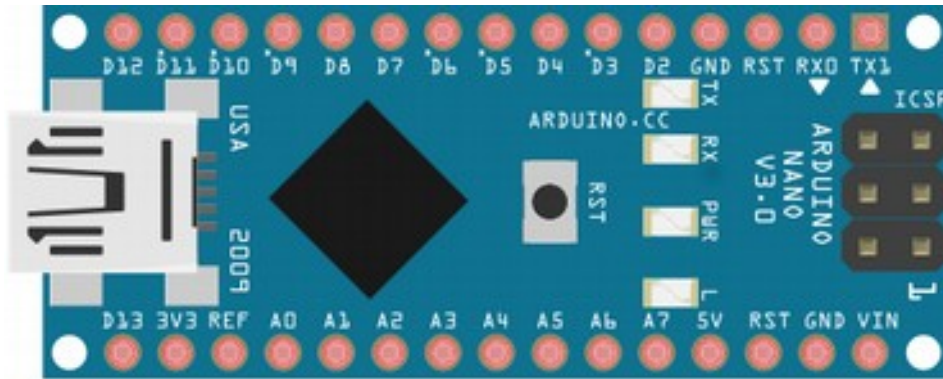


This image was originally posted to
Flickr by mbanzi at
<https://flickr.com/photos/83986067@N00/5537363651>

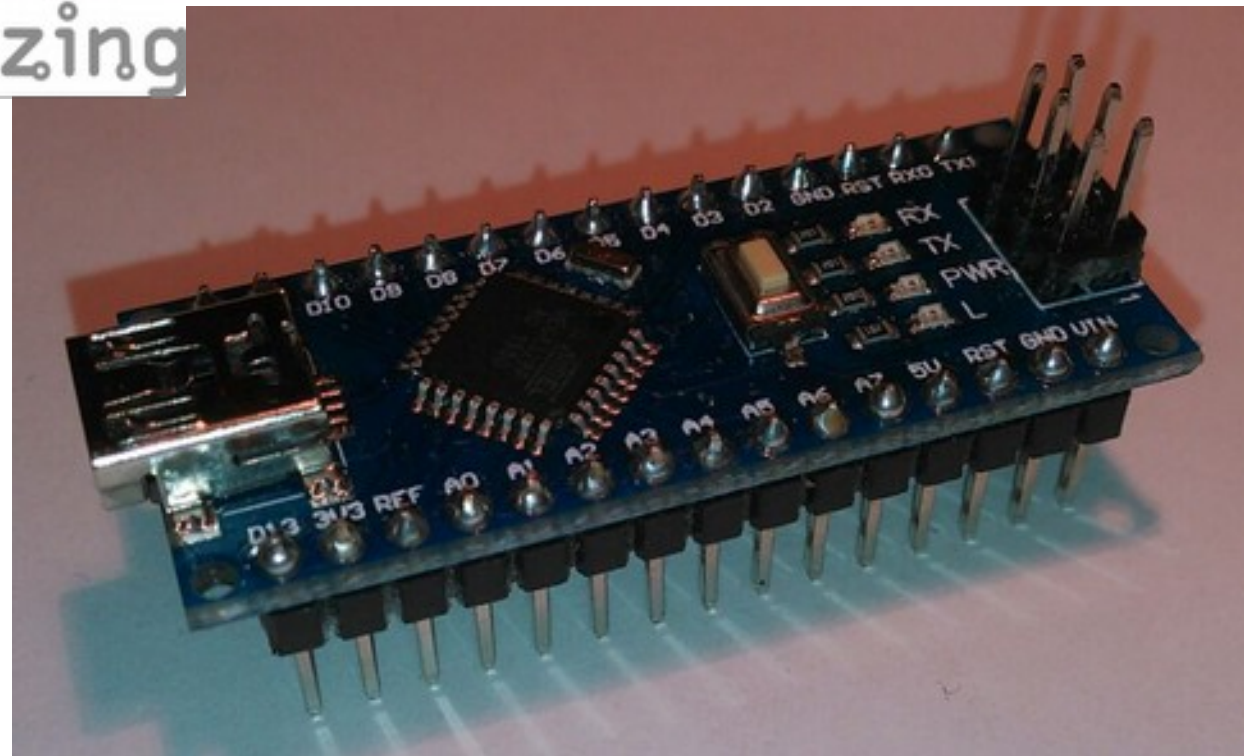
. It was reviewed on 16 October 2016
by FlickreviewR and was confirmed to
be licensed under the terms of the cc-
by-sa-2.0.



Il nostro SUPER elaboratore (Arduino NANO compatibile)

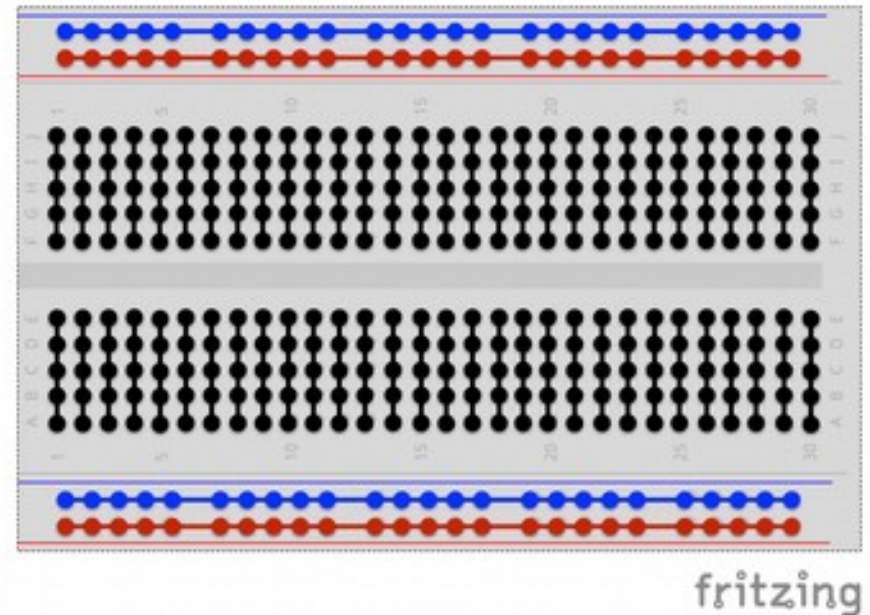
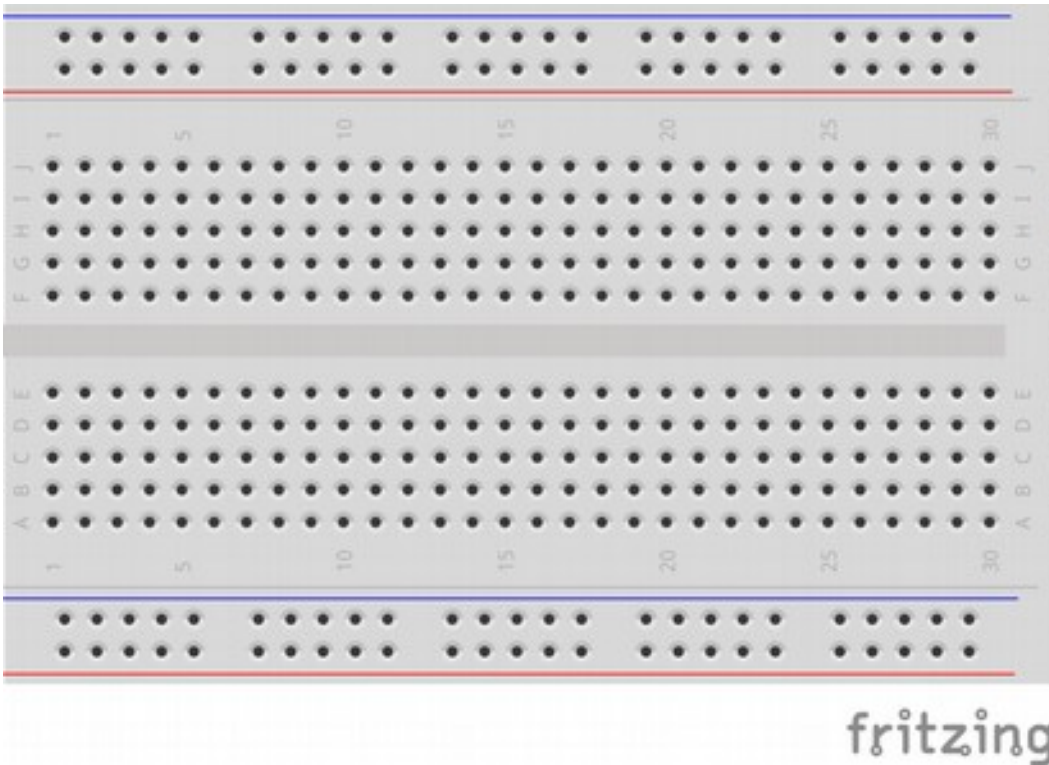


fritzing

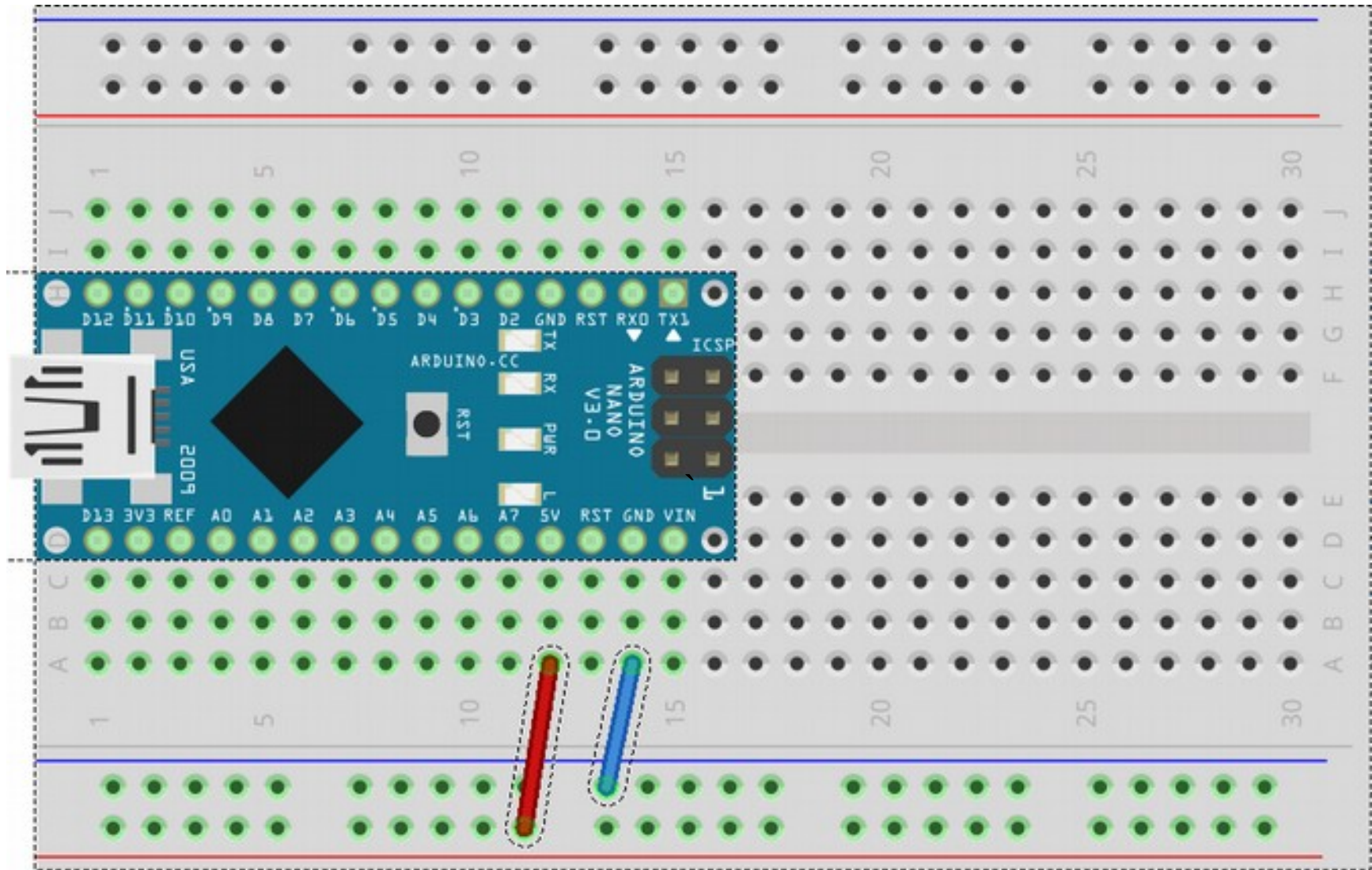


BREADBOARD

la tavoletta di pane (ma non è commestibile)



Il nostro SUPER elaboratore NANO + breadboard

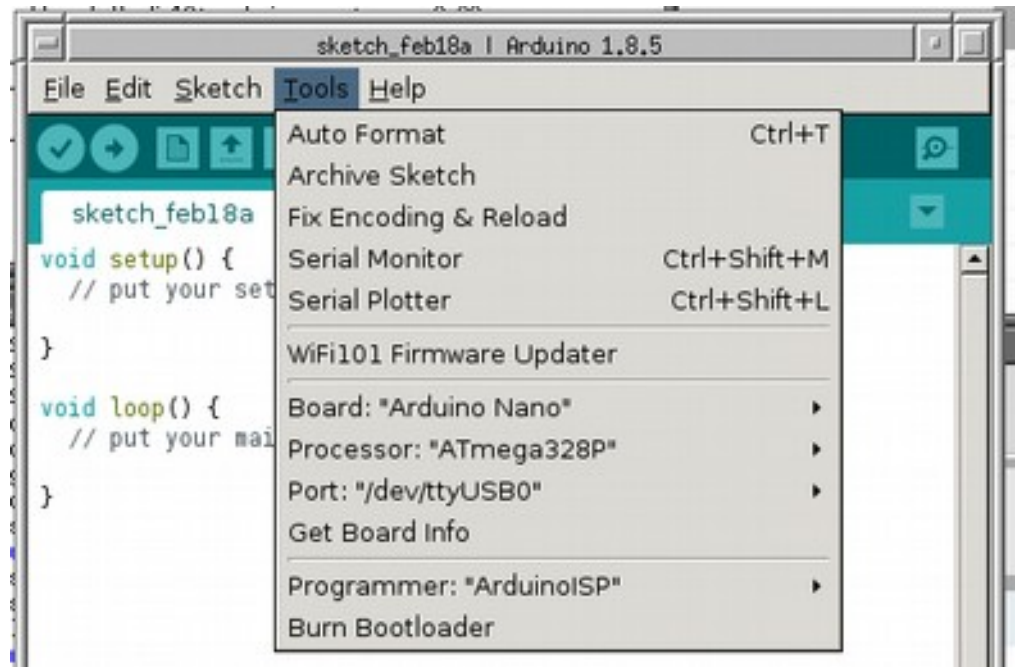


fritzing



Esercizio Zero non fare nulla :)

- Ma lo facciamo fare al NANO
- Attacchiamo il NANO alla porta USB
- Nell'IDE scegliamo la scheda giusta:
 - Nel MENU Tools deve comparire:



Lo sketch che non fa nulla:

```
void setup() {
```

```
  // put your setup code here, to run once:
```

```
}
```

```
void loop() {
```

```
  // put your main code here, to run repeatedly:
```

```
}
```



GO!



Cosa è successo

- ????



Esercizio 1: blinka blinka led!

- Ora proviamo un esercizio che faccia qualcosa di minimale....
- Facciamo lampeggiare un led del NANO (quello contrassegnato con la lettera L).
- Scegliamo l'esempio dal menu
 - File/Examples/01.Basics/Blink



Ecco il codice dello sketch

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);                      // wait for a second
}
```



GO!



Idee...

- Provate a cambiare la frequenza e il pattern acceso/spento del lampeggio (modificate i valori dei delay) per vedere cosa succede
- Arduino è autonomo. Se provate a staccare la spina USB dal PC e attaccate Arduino a un semplice alimentatore continua a eseguire il programma che avete caricato.



NOTA IMPORTANTE

- STIAMO PER COSTRUIRE CIRCUITI ELETTRONICI
- Staccate la spina USB dal COMPUTER
- Inserite i componenti e i ponticelli nella breadboard
- PRIMA DI RICOLLEGARE LA SPINA USB fate controllare il circuito a un mentor
- ... solo dopo il controllo ricollegate l'Arduino.



Componenti: la resistenza

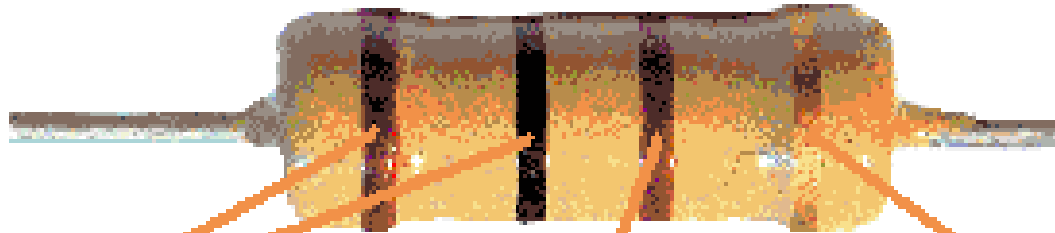
- Serve per limitare la corrente (la corrente elettrica fa *fatica* ad attraversare una resistenza)
- La *fatica* si misura in OHM
 - Si indica con la lettera Ω
- + OHM – Corrente
- Il valore si legge dalle strisce colorate
- Quella qui illustrata è da 100Ω (marrone – nero – marrone – oro).



Simbolo grafico della
resistenza elettrica



Il codice colore delle resistenze



Valori numerici

Nero	0
Marrone	1
Rosso	2
Arancio	3
Giallo	4
Verde	5
Blu	6
Viola	7
Grigio	8
Bianco	9

Moltiplicatore

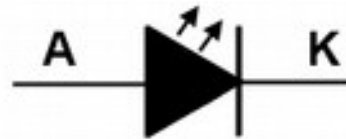
Argento	0,01
Oro	0,1
Nero	1
Marrone	10
Rosso	100
Arancio	1K
Giallo	10K
Verde	100K
Blu	1M
Viola	10M

Tolleranza

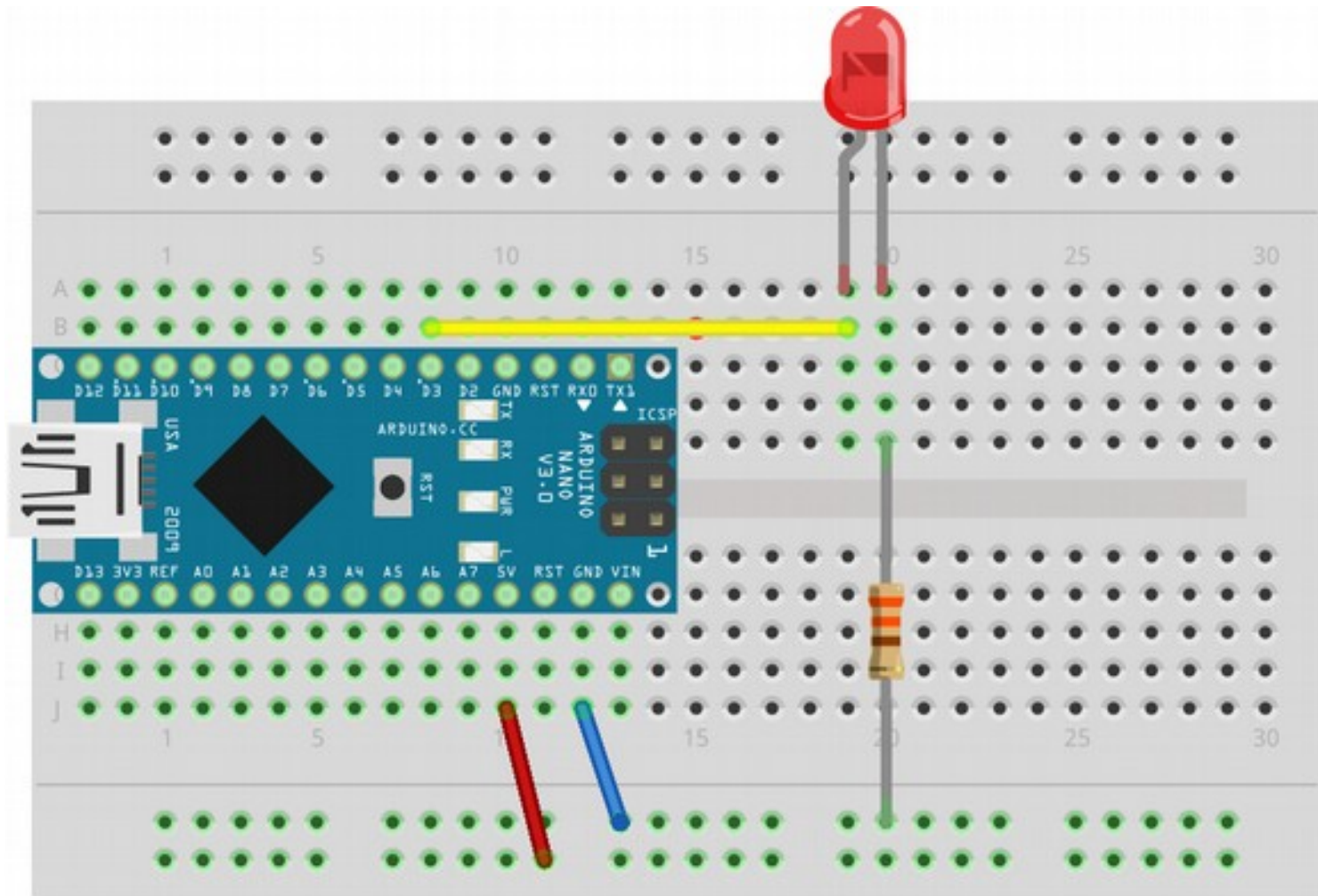
Argento	±10%
Oro	±5%
Marrone	±1%
Rosso	±2%
Verde	±0,5%
Blu	±0,25%
Viola	±0,1%

Il diodo LED

- Light Emitting Diode
- È un diodo semiconduttore:
 - Per accendersi la corrente deve fluire dall'anodo al catodo
 - Regola pratica (thumb rule: il filo + lungo va collegato al polo positivo +)
 - **ATTENZIONE:** devono passare solo 10mA. Senza opportuna resistenza si rompono!



Esercizio 2: il nostro primo circuito



fritzing



Facciamo lampeggiare il led rosso

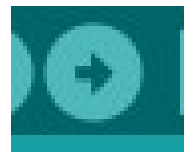
- **Basta modificare il codice (sostituendo 3 a LED_BUILTIN)**

```
// the setup function runs once when you press reset or power the board
```

```
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(3, OUTPUT);  
}
```

```
// the loop function runs over and over again forever
```

```
void loop() {  
  digitalWrite(3, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);           // wait for a second  
  digitalWrite(3, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);           // wait for a second  
}
```



IDEE per altri esperimenti

- Facciamo lampeggiare sia il led rosso sia il led piccolo sul NANO in modo alternato (quando è acceso uno l'altro è spento)

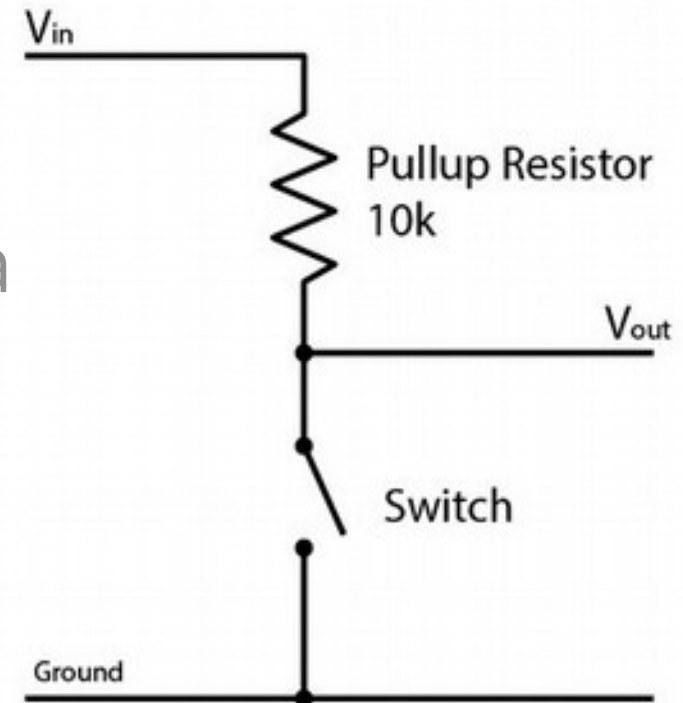
```
void setup() {  
  pinMode(3, OUTPUT);  
  pinMode(LED_BUILTIN, OUTPUT);  
}
```

```
void loop() {  
  digitalWrite(3, HIGH);  
  digitalWrite(LED_BUILTIN, LOW);  
  Delay(1000);  
  digitalWrite(3, LOW);  
  digitalWrite(LED_BUILTIN, HIGH);  
  Delay(1000);  
}
```



Esercizio 3: leggiamo un input

- Invece di un pulsante useremo un ponticello.
- Occorrerebbe una resistenza per configurare il valore di *default*.
- Il processore del nostro NANO ha già all'interno la resistenza di PULLUP (valore default = 1) basta configurarla via software



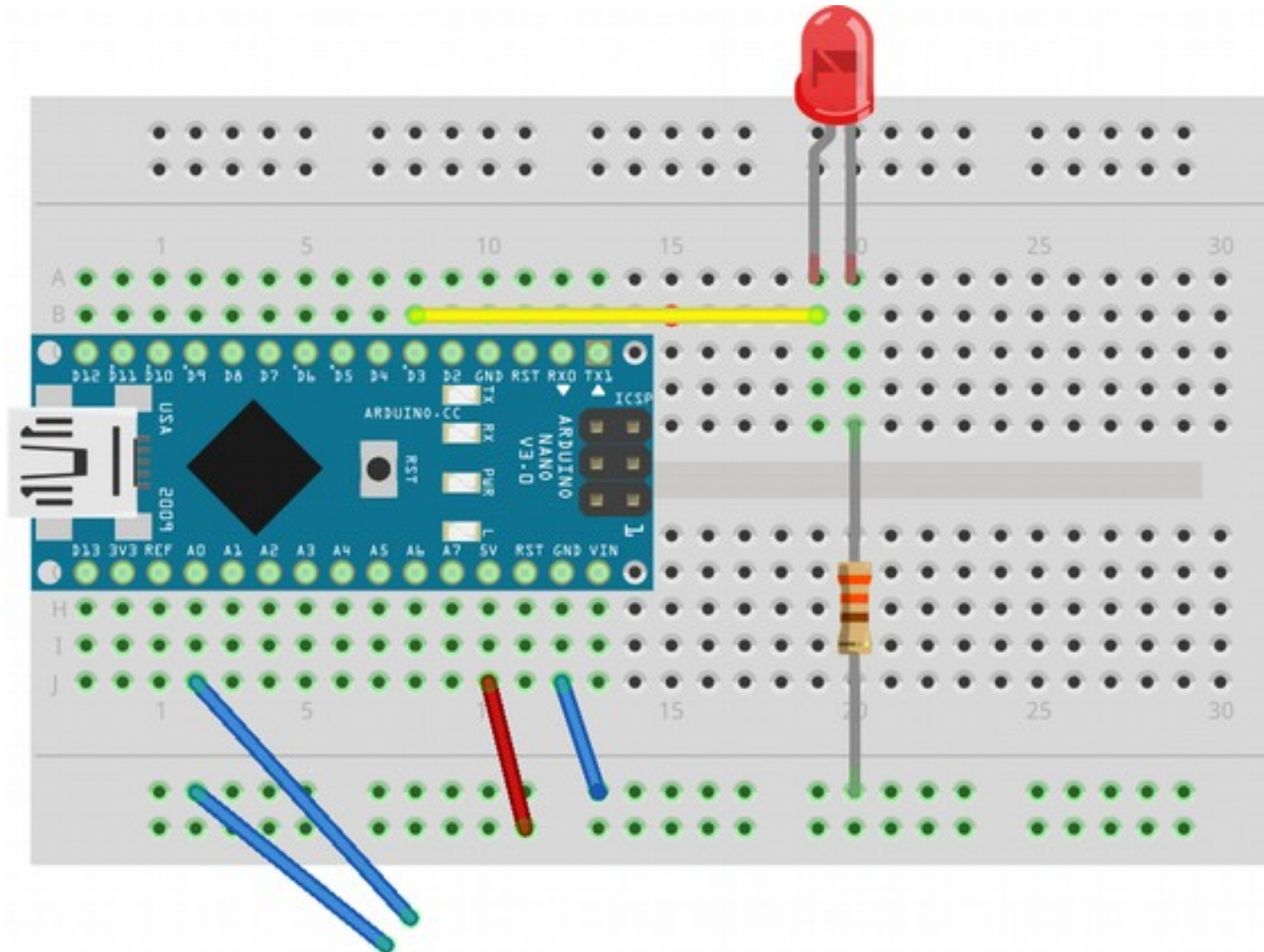
Ecco lo sketch per l'input

```
void setup() {  
  pinMode(3, OUTPUT);  
  pinMode(A0, INPUT_PULLUP);  
}
```

```
void loop() {  
  if (digitalRead(A0) == LOW)  
    digitalWrite(3, HIGH);  
  else  
    digitalWrite(3, LOW);  
}
```



Ponticello per input



fritzing



Ecco lo sketch dell'interruttore per le luci delle scale

- Basta cambiare poche righe:

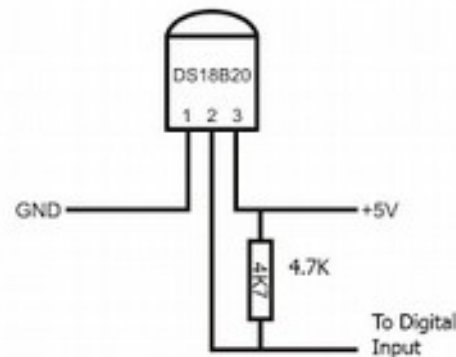
```
void setup() {  
  pinMode(3, OUTPUT);  
  pinMode(A0, INPUT_PULLUP);  
}
```

```
void loop() {  
  if (digitalRead(A0) == LOW) {  
    digitalWrite(3, HIGH);  
    delay(10000);  
    digitalWrite(3, LOW);  
  }  
}
```

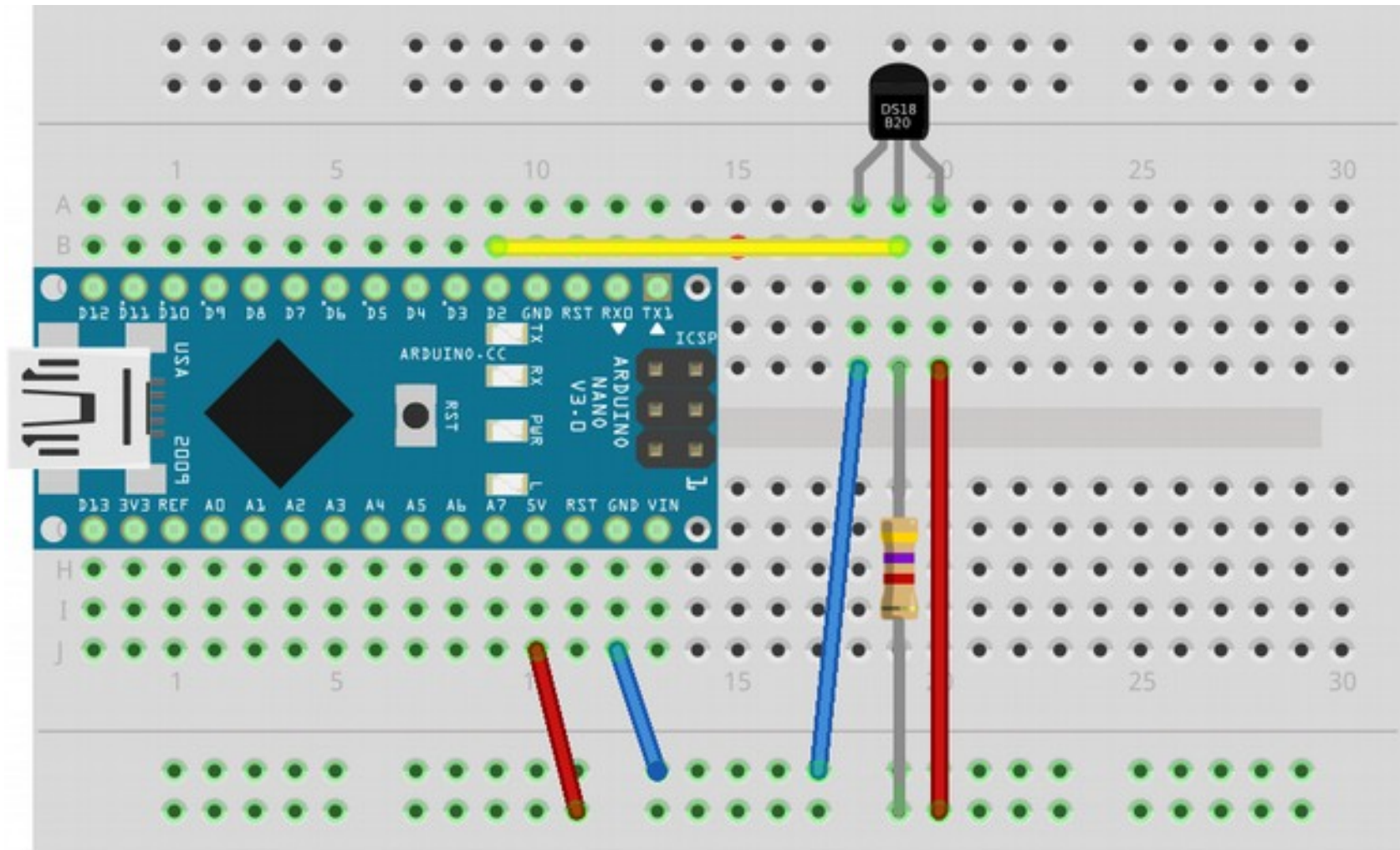


Esercizio 4: leggiamo la temperatura

- Occorre un sensore.
 - Useremo un componente molto intelligente (di fatto è un elaboratorino anche lui!)
 - Si chiama DS18B20
- Occorrono funzioni per semplificare il dialogo con questo componente
 - Queste collezioni di funzioni si chiamano librerie



Il circuito per il DS18B20



fritzing



Aggiungiamo la libreria

- Selezionare la voce di menu
Sketch/Include Library/Manage Libraries...
- Cercare “DallasTemperature”
Premere il bottone INSTALL



La libreria viene fornita completa di esempi

- Proviamo l'esempio "SIMPLE":
 - Scegliamo il menu:
 - File/Examples/DallasTemperature/Simple
- Prima di attivare lo sketch aprite la console seriale
 - Scegliamo il menu:
 - Tools/Serial Monitor



Simple

```
// Include the libraries we need
#include <OneWire.h>
#include <DallasTemperature.h>
// Data wire is plugged into port 2 on the Arduino
#define ONE_WIRE_BUS 2
// Setup a oneWire instance to communicate with any OneWire devices
OneWire oneWire(ONE_WIRE_BUS);
// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);
* The setup function. We only start the sensors here *
void setup(void) {
  Serial.begin(9600);
  Serial.println("Dallas Temperature IC Control Library Demo");
  sensors.begin();
}
/* Main function, get and show the temperature */
void loop(void)
{
  // call sensors.requestTemperatures() to issue a global temperature
  Serial.print("Requesting temperatures...");
  sensors.requestTemperatures(); // Send the command to get temperatures
  Serial.println("DONE");
  // We use the function ByIndex, and as an example get the temperature from the first sensor only.
  Serial.print("Temperature for the device 1 (index 0) is: ");
  Serial.println(sensors.getTempCByIndex(0));
}
```



Ma funziona?

- Provate a toccare il DS18B10
 - La temperatura dovrebbe salire
- Ora fate vento con un foglio
 - La temperatura dovrebbe scendere...

```
Requesting temperatures...DONE
Temperature for the device 1 (index 0) is: 20.50
Requesting temperatures...DONE
Temperature for the device 1 (index 0) is: 20.87
...
Temperature for the device 1 (index 0) is: 22.00
Requesting temperatures...DONE
Temperature for the device 1 (index 0) is: 22.06
Requesting temperatures...DONE
Temperature for the device 1 (index 0) is: 22.12
Requesting temperatures...DONE
Temperature for the device 1 (index 0) is: 22.25
...
Requesting temperatures...DONE
Temperature for the device 1 (index 0) is: 22.06
Requesting temperatures...DONE
Temperature for the device 1 (index 0) is: 22.00
Requesting temperatures...DONE
Temperature for the device 1 (index 0) is: 21.56
Requesting temperatures...DONE
Temperature for the device 1 (index 0) is: 21.19
Requesting temperatures...DONE
Temperature for the device 1 (index 0) is: 20.87
Requesting temperatures...DONE
Temperature for the device 1 (index 0) is: 20.81
Requesting temperatures...DONE
```



Per fare un termostato...

- **Basta cambiare qualche riga di codice:**
- **Nel setup aggiungiamo**

```
pinMode(LED_BUILTIN, OUTPUT);
```

- **E la funzione LOOP diventa:**

```
void loop(void)
{
  float temp;
  sensors.requestTemperatures();
  Serial.print("Temperature for the device 1 (index 0) is: ");
  temp = sensors.getTempCByIndex(0);
  Serial.println(temp);
  if (temp > 22)
    digitalWrite(LED_BUILTIN, HIGH);
  else
    digitalWrite(LED_BUILTIN, LOW);
}
```



Cosa succede?

- Il LED si accende se la temperatura sale oltre i 22 gradi e si spegne se scende al di sotto. (o altro valore basta cambiare la costante)
- È il circuito che può essere usato per attivare il condizionatore quando è troppo caldo
- ... e se invece fosse un impianto di riscaldamento? Occorre invertire accendere la caldaia (il led) quando è freddo e spegnere quando è caldo. Come si modifica il codice?



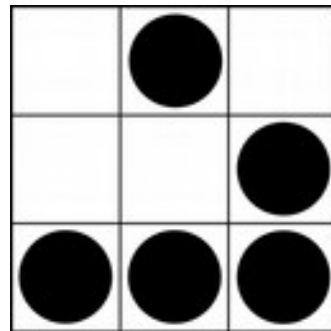
È solo l'inizio...

- Abbiamo visto insieme solo alcuni esempi di making.
- Cercando sul web consigli e facendo tanti esperimenti si può dare sfogo alla propria fantasia per creare cose digitali.
- Gli strumenti usati (circuiti, linguaggi) sono **veri**: non sono giocattoli per bimbi.
- I componenti elettronici sono gli stessi usati nei circuiti dei dispositivi prodotti dalle industrie, i linguaggi (C, C++, Python, Shell) sono quelli usati per scrivere Sistemi Operativi, Applicazioni e nelle più importanti aziende che operano nel mondo digitale.
- Siamo in tanti che stiamo facendo cose serie giocando.



We are still creating art and beauty on a
computer:

the art and beauty of revolutionary ideas
translated into (libre) code...



renzo, rd235, iz4dje